AFRL-RI-RS-TN-2013-001

SECURITY TAGGED ARCHITECTURE CO-DESIGN (STACD)

MARCH 2013

INTERIM TECHNICAL NOTE

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TN-2013-001 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /

STEVEN T. JOHNS, Chief
Trusted Systems Branch

/ S /

RICHARD MICHALAK,
Acting Technical Advisor, Computing &
Communications Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* MARCH 2013	**2. REPORT TYPE** INTERIM TECHNICAL NOTE	**3. DATES COVERED** *(From - To)* JAN 2011 – DEC 2012

4. TITLE AND SUBTITLE Security Tagged Architecture Co-Design (STACD)	**5a. CONTRACT NUMBER** IN-HOUSE
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER 62788F
6. AUTHOR(S) Jonathan Heiner	**5d. PROJECT NUMBER** T2ST
	5e. TASK NUMBER IN
	5f. WORK UNIT NUMBER HO
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/Information Directorate Rome Research Site/RITA 525 Brooks Road Rome NY 13441-4505	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/Information Directorate Rome Research Site/RITA 525 Brooks Road Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RI
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TN-2013-001

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# 88ABW-2013-1318
Date Cleared: 20 March 2013

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Security Tagged Architecture Co-Design (STACD) initiative focuses on eliminating inherent software vulnerabilities by redesigning the underlying hardware and the operating system to enforce software security policies and semantics. The new approach will use a metadata processing unit known as the tagged management unit (TMU) that operates concurrently with the CPU to process the metadata. The introduction of tag capable hardware requires software that uses tagged information. We will develop a tag enabled Operating System (OS) that permits the simplification and reduction in size of the OS for easier verification and validation. The STACD project will co-design a new scalable Security Tagged Multicore Processor (STMP), a Security Tagged Zero-Kernel OS (ST-ZKOS), and a Security Tagged Interconnect (STI) that will maintain metadata through execution without negatively influencing performance by processing the data and its corresponding metadata in parallel. This system will enforce software semantics and security policies, guarantee isolation and separation of information, and provide resistance to malicious attacks. The co-design approach provides a higher assurance of compatibility between the components and a stronger security base.

15. SUBJECT TERMS

Multicore Architecture, Manycore Architecture, Security, Zero-kernel OS, Tagging

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **JONATHAN HEINER**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	11	**19b. TELEPONE NUMBER (*Include area code)*** **N/A**

TECHNICAL NOTE

Security Tagged Architecture Co-Design (STACD) Technical Status Report
As of 31-Dec-2012

Effort Objective:

The Security Tagged Architecture Co-Design (STACD) initiative focuses on eliminating inherent software vulnerabilities by redesigning the underlying hardware and the operating system to enforce software security policies and semantics.

The new approach will use a metadata processing unit known as the tagged management unit (TMU) that operates concurrently with the CPU to process the metadata. The introduction of tag capable hardware requires software that uses tagged information. We will develop a tag enabled Operating System (OS) that permits the simplification and reduction in size of the OS for easier verification and validation.

The STACD project will co-design a new scalable Security Tagged Multicore Processor (STMP), a Security Tagged Zero-Kernel OS (ST-ZKOS), and a Security Tagged Interconnect (STI) that will maintain metadata through execution without negatively influencing performance by processing the data and its corresponding metadata in parallel. This system will enforce software semantics and security policies, guarantee isolation and separation of information, and provide resistance to malicious attacks. The co-design approach provides a higher assurance of compatibility between the components and a stronger security base.

Effort Metrics and Logistics:

Tagged Secure Multicore Processor:

Performance Metric: A 2 to 3X performance degradation using tagging to implement Dynamic Instruction Flow Tracking (DIFT) compared to normal operation. Previous DIFT techniques incurred a performance degradation of 3.6X to 37X. < 10% performance degradation compared to software running on CPU alone.

Area Metric: <20% increase in area compared to Core. No comparable data from previous techniques due to novel approach.

Security Metric: 100% detection/prevention of half of 2011 CWE/SANS Top 25 Most Dangerous Software Errors.

Completeness Criteria: Maintains correctness of tags through execution. Determined through analysis of input and output compared to a golden copy.

Tagged Secure Zero Kernel OS:

Performance Metric: >25% of reduction of context switches. No comparable data from previous techniques due to novel approach.

Size Metric: 10% less LoC (lines of Code) compared to similar OS, 10% fewer trusted API instructions than similar OS.

Security Metric: 100% detection/prevention of half of 2011 CWE/SANS Top 25 Most Dangerous Software Errors.

Completeness Criteria: Functioning OS, i.e. correct execution and separation of applications, capable of interfacing with Tagged hardware.

Tagged Interconnect:

Performance Metric: <10% performance degradation through interconnect. No comparable data from previous techniques due to novel approach.

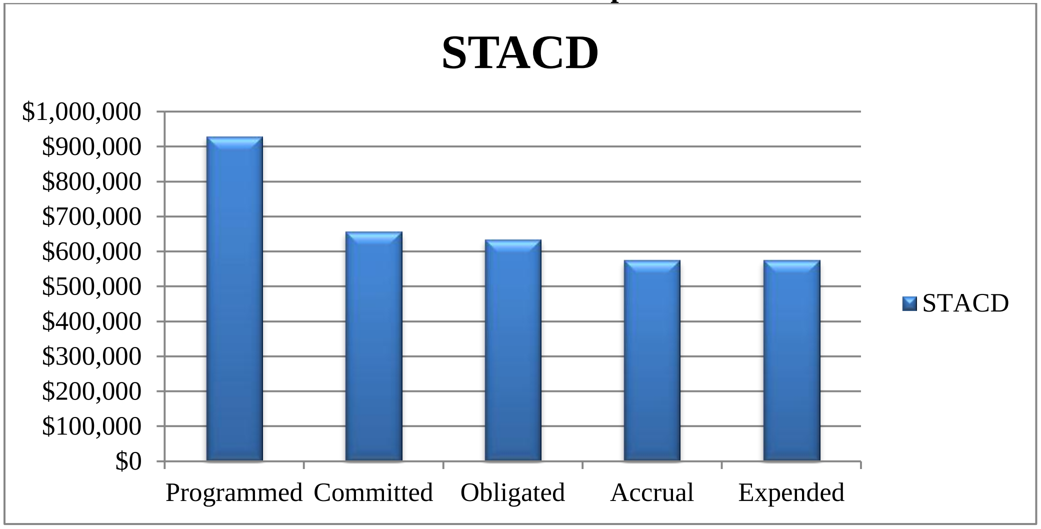
Area Metric: < 100% increase in area compared to original interconnect system.

Completeness Criteria: Correct transfer of data and associated tag information. Verified through simulation and testing.

Financial Status

All financial projections have been met to date. No modifications needed.

FY13 Financial Snapshot



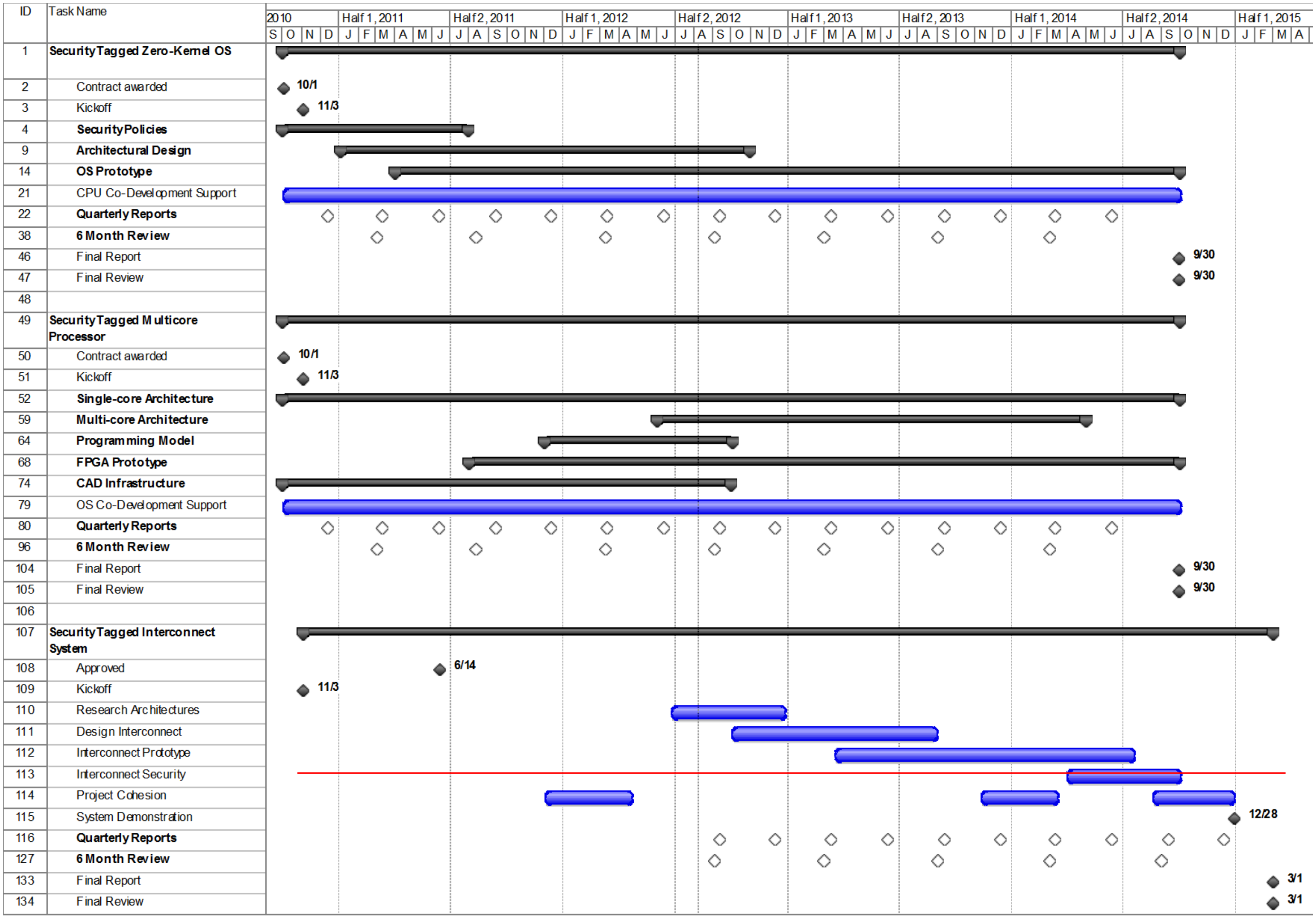
Accrual: contractor has billed DFAS. Expended: DFAS has paid the contractor.

Original Project Timeline

	‘10		2011												2012												2013										2014						
Key Tasks	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Security Tagged ZKOS																																											
Security Policies	■	■	■	■	■	■	■																																				
Architectural Design			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																					
OS Prototype					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■							
Security Tagged Multicore Processor																																											
Single core Architecture	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																■	■	■	■							
Multicore Architecture																						■	■	■	■	■	■	■	■	■	■	■	■	■									
Programming Model																■	■	■	■	■	■	■	■	■																			
FPGA Prototype												■	■	■	■	■	■	■	■	■	■									■	■	■	■	■	■	■							
CAD Infrastructure	■	■	■	■	■	■	■	■	■							■	■																										
Security Tagged Interconnect																																											
Research Architectures					■	■	■	■	■	■	■																																
Design Interconnect											■	■	■	■	■	■	■	■	■	■																							
Interconnect Prototype																			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■									
Interconnect Security																																	■	■	■	■	■	■					
Project Cohesion																																					■	■	■	■	■	■	■

The Security Tagged Interconnect System’s long schedule time is caused by dependence upon the supporting contracts, Security Tagged Multicore Processor and Security Tagged ZKOS. Due to budget constraints, the supporting contracts have extended into FY14, and will provide half man-year efforts both years. This will also extend the Security Tagged Interconnect schedule into FY15. An updated schedule is provided below.

New Project Timeline



ID	Task Name
1	**Security Tagged Zero-Kernel OS**
2	Contract awarded
3	Kickoff
4	**Security Policies**
9	**Architectural Design**
14	**OS Prototype**
21	CPU Co-Development Support
22	**Quarterly Reports**
38	**6 Month Review**
46	Final Report
47	Final Review
48	
49	**Security Tagged Multicore Processor**
50	Contract awarded
51	Kickoff
52	**Single-core Architecture**
59	**Multi-core Architecture**
64	**Programming Model**
68	**FPGA Prototype**
74	**CAD Infrastructure**
79	OS Co-Development Support
80	**Quarterly Reports**
96	**6 Month Review**
104	Final Report
105	Final Review
106	
107	**Security Tagged Interconnect System**
108	Approved
109	Kickoff
110	Research Architectures
111	Design Interconnect
112	Interconnect Prototype
113	Interconnect Security
114	Project Cohesion
115	System Demonstration
116	**Quarterly Reports**
127	**6 Month Review**
133	Final Report
134	Final Review

Timeline changes

No changes to timeline except those as stated above.

Progress Towards Planned Objectives (per subtask) as of Dec. 31 2012

Security Tagged ZKOS (University of Idaho)

Security Policies (Task 1): The first task investigates the set of security policies that are enforceable by Security Tagged Architecture (STA). Prior work in this area defines run-time solutions as meeting a set of enforceable security polices, which are a subset of all policies. It remains an open question if an STA can support the set of all run-time enforceable security policies. We contend that all executable hardware of the system (DMA controllers, co-processors, network cards, etc.) will have to conform to STA principles, or will have to be isolated by STA hardware in order to provide strong assurance of run-time enforceable security policies.

Task 1.1 Map the set of security policies of the TMU of the Trust Management, Intrusion-tolerance, Accountability, and Reconstitution Architecture (TIARA) STA to the set of run-time enforceable security policies as defined by Hamlen, Morrisett and Schneider. Formally specify the relationship of the STA policies to the formal classes.

We completed the evaluation but delayed the mapping of STA policies to the formal classes until after we have completed development of the STA policies.

Task 1.2 Document new hardware features that are needed to expand the set of security policies enforceable by the proposed STA.

- Comments: This is an ongoing process with Cornell. We completed the initial work and discussions in 2011. At this point, we are working with Cornell as they finalize their hardware designs. For example, our security policy requires the use of a “copy-bit” in the hardware. This is appearing to have a higher impact than we originally suspected.

  For this period, we planned to expand this work to look at hardware support for high-level semantic attacks. The literature suggests that this has been successful, but we question the context of the claims. Are we just pushing the problem off? Or are there additional hardware features that can be used to support protection from high-level semantic attacks (or at least those that result in authentication and access-control bypass?)

- Objectives: Regular discussion with Cornell and analysis of semantic level attacks.
- Progress: Regular discussions with Cornel; literature review of semantic level attacks. We are now documenting that review.

Task 1.3 Evaluate the limitations of a processor-only STA compared to a system-wide STA. This evaluation will include an impact analysis on the effectiveness of the STA and proposed mitigations to negative impacts.

- Comments: We completed this work for the first phase of this project. We are now working with Cornell on continued discussions and revisions of the work.
- Objectives: Regular discussion with Cornell; begin design of multi-core simulator model.
- Progress: Regular discussions with Cornell. Worked on multi-core simulator model and design. Began design and implementation.

Architectural Design (Task 2): The second task is the development of an architectural design of a new class of operating system that utilizes security enforcement mechanisms of tagged-architectures to implement a MILS compliant system.

Task 2.1 Identification of key hardware features that are necessary for the efficient implementation of a new operating system architecture.

We completed this work for the first phase of this project. We are now working with Cornell on continued discussions and revisions of the work. Completed a draft summary document of the current “tagging scheme”, which designs these hardware features.

Task 2.2 Develop high level design of new operating system by exploring the features developed through prior AFRL research and expand upon those which give the most support to an MILS solution.

- Comments: We changed this work to include an extension of the project involving modifications and changes to the RTEMS operating system.

- Objectives: Extend design modifications to include support for user level code libraries; continue testing and refinement of the modifications. Document suite of test cases, and evaluate completeness of test suite. Currently the test suite runs under the simulator; we may need to make modifications to run on the hardware when we get it.
- Progress: Continued testing and refinement of the modifications. Document suite of test cases, and evaluate completeness of test suite. Currently the test suite runs under the simulator; we may need to make modifications to run on the hardware when we get it.

Task 2.3 Analyze the proposed architectural solution with respect to the set of run-time enforceable security policies and compliance with the MILS architecture.

- Objectives: Conduct this evaluation.
- Progress: This evaluation is being conducted in conjunction with the test cases mentioned in Task 2.2 and the modification of the architecture. We have made partial progress, but need to complete the modifications and tests cases to complete the evaluation.

OS Prototype (Task 3): The third task involves the development of a prototype operating system that implements the design of the second phase and executes on a tagged-architecture processor.

Task 3.1 Determine target architecture for the implementation. If a simulator or implementation of such architecture is not readily available through AFRL channels, we will propose a mechanism for simulating the base hardware for our new OS. This task, as stated, is complete. However we are moving forward with implementation of a testbed under this task.

- Comments: It was originally determined to use an FPGA-based system as the testbed. This is up and running, but requires Cornell hardware (HW) changes for evaluation. To speed up the process we decided to also develop software simulators in support of the Tagging.
- Objectives: Continue improvement of simulators and deploy any versions of HW we receive from Cornell.
- Progress: We have the initial tagging scheme completely implemented in our simulator and are using it to run the test cases mentioned previously. We also have a simulator platform for evaluating the DIFT-style tagging scheme which will be an addition to our tagging scheme. We have begun implementing an additional DIFT-style scheme to evaluate effectiveness of these schemes. We have not received any HW from Cornell, so that part to of the work is still delayed. Received new hardware model Dec 22nd.

Task 3.2 Implement core functionality of the operating system to enable execution of middleware and user level processes to demonstrate security features.

- Comments: This task was modified to involve the modification of the RTEMS code base.
- Objectives: Complete testing of modifications using simulator and Cornell HW.
- Progress: We completed the modifications of the RTEMS code to support security tagging. The evaluation of these modifications are undergoing test in the simulator.

Task 3.3 Enhance implementation in support of key features defined in consultation with AFRL personnel.

- Objectives: None planned
- Progress: None planned

Security Tagged Multicore Processor (Cornell University)

Single-core Architecture

Objectives: Investigate mechanisms for precise exceptions. Investigate mechanisms for efficiently managing large tags. More specifically, collect statistics to see how much spatial and temporal value locality exists in typical tagging techniques. Develop ways to predict impact of tagging on real-time systems. Develop mechanisms for high-performance tagging: new tag processing units and efficient management of large tags. Evaluate the tag compression technique and the tagging scheme for real-time systems.

Progress: We have looked into design options to support precise exceptions on in-order processors, and started implementing the techniques. Experimental results on dynamic taint analysis show that tags for contiguous memory regions often have the same value (spatial value locality) and results for array bound checks do not show high spatial locality but show good temporal locality. We designed a compressed tag cache to efficiently store large tags on-chip. A preliminary analysis indicates that the compression scheme can effectively increase the cache size by a factor of two to four for dynamic information flow tracking and array bound checking. We also came up with an initial formulation to compute the worst-case execution time (WCET) of a program including the impact of tagging. We also started investigating techniques to improve performance of flexible tagging. We investigated new tag processing units that can still support a range of tagging techniques, but with high performance (a few GHz operating frequency). We finished an initial design and evaluation of a fixed datapath. The design has been prototyped in register-transfer level (RTL) and evaluated – the result has been submitted and accepted to the International Conference on Dependable Systems and Networks (DSN) 2012. We designed a tag cache compression technique and also designed a tagging scheme that performs checks while guaranteeing real-time constraints. We started working on simulation studies to evaluate both schemes. We have continued working on simulators and benchmarks.

Multi-core Architecture

Objectives: Setup a multi-core processor - unmodified Leon3 processors.

Progress: A student finished setting up the unmodified Leon3 in a dual-core configuration on an FPGA.

Programming Model

Objectives: Investigate implementing tagging schemes using a high-level synthesis tool.

Progress: We have started investigating this direction. A student successfully set up a high-level synthesis tool, named C-to-Silicon from Cadence that allows writing a hardware design in SystemC. Implemented a few tagging schemes in SystemC and compared the results with implementations in RTL. The paper on a framework and high-level synthesis of tagging hardware was accepted to International Conference on Computer Design (ICCD). We generated the final version of the paper.

FPGA Prototype

Objectives: Implement a tagging scheme (dynamic taint analysis) on an FPGA. Implement the new tagging scheme from the OS group in RTL.

Progress: We have worked on implementing the FlexCore tagged architecture on an FPGA. However, the tag memory system needs to be further debugged. In order to help the prototype efforts, we recruited two new students in September and one in October. The students learned our architecture and RTL code. They started implementing precise exception mechanisms and debugging tag memory systems. In order to facilitate collaboration with the OS team, we also sent our current design and had it run at the University of Idaho. We have successfully debugged all modules and have a simple DIFT demo working in simulations. We are working on implementing new interface instructions and also debugging the design for an FPGA board. We have worked on implementing the FlexCore tagged architecture on an FPGA. We finished debugging the FPGA prototype for common functions and had a demo at the DARPA CRASH meeting. We also started implementing the new tagging scheme. We have finished the RTL coding and are working on debugging. We finished the RTL coding, tested the design, and wrote a document on the prototype. The prototype runs on an FPGA.

CAD Infrastructure

Objectives: Finish the VPR flow and try out different FPGA architectures.

Progress: We started looking into setting up an open-source FPGA tool flow named VPR. Assigned a student to this task. We have set up the VPR flow and evaluated exiting tagging designs on different FPGA architectures.

Security Tagged Interconnect (In-House)

Research Architectures

Finished reading LEON3 documentation and AMBA bus protocols.

Design Interconnect

Have identified technique to cause all DMA-based on-chip peripherals to retrieve a tag from a list of available tags and store the tag in its header information for further use; then whenever the DMA-based on-chip peripheral reads/writes content to memory/CPU the correct associated tag is forwarded as well. Each on-chip peripheral will be required to be associated with only a single user.

Interconnect Prototype

No progress to report.

Interconnect Security

No progress to report.

Project Cohesion

Close collaboration between the University of Idaho and Cornell University will allow seamless integration of components. Currently, the STMP prototyped by Cornell University implements the security policies defined by the University of Idaho‘s ST-ZKOS. Cornell University has shared the prototype code with AFRL/RITA and the University of Idaho.

Issues or Concerns

No issues or concerns to date.

Deliverables

- Monthly status reports from University of Idaho (ST-ZKOS) and Cornell University(TSMP).
- TSMP Single-core Prototype, Nov. 2011.
- TSMP Multi-core Prototype, Oct. 2014.
- TS-ZKOS Prototype, Oct. 2014.
- TI Prototype, Oct. 2014.

Publications/Invention Disclosures/Patents/…

- No in-house sourced as of Dec. 31 2012.
- Daniel Y. Deng and G. Edward Suh, "Precise Exception Support for Decoupled Run-Time Monitoring Architectures", Proceedings of the 29th International Conference on Computer Design (ICCD), September 2011.
- Daniel Y. Deng, and G. Edward Suh, "High-Performance Parallel Accelerator for Flexible and Efficient Instruction-Grained Run-Time Monitoring", to appear in Proceedings of Dependable Systems and Networks (DSN), June 2012.
- Daniel Lo, and G. Edward Suh, "Worst-Case Execution Time Analysis for Parallel Run-Time Monitoring", to appear in Proceedings of Design Automation Conference (DAC), May 2012.
- J. Song and J. Alves-Foss, "Security Tagging for a Zero-Kernel Operating System", to appear in HICCS, Jan. 2013.
- Mohamed Ismail and G. Edward Suh, "Fast Development of Hardware-Based Run-Time Monitors Through Architecture Framework and High-Level Synthesis", Proceedings of the International Conference on Computer Design (ICCD), October 2012.

Planned Future Activities

Security Tagged ZKOS

Security Policies (Task 1): The first task investigates the set of security policies that are enforceable by an STA. Prior work in this area defines run-time solutions as meeting a set of enforceable security polices, which are a subset of all policies. It remains an open question if an STA can support the set of all run-time enforceable security policies. We contend that all executable hardware of the system (DMA controllers, co-processors, network cards, etc.) will have to conform to STA principles, or will have to be isolated by STA hardware in order to provide strong assurance of run-time enforceable security policies.

Task 1.1 No work on this task this period.

Task 1.2 Continue documentation of review and propose solution to problem.

Task 1.3 Continue discussions with Cornell. Complete multi-core simulator model, design and implementation.

Architectural Design (Task 2): The second task is the development of an architectural design of a new class of operating system that utilizes security enforcement mechanisms of tagged-architectures to implement a MILS compliant system.

Task 2.1 Compare design with hardware model and make "corrections".

Task 2.2 Continue testing and refinement of the modifications. Document suite of test cases, and evaluate completeness of test suite. Currently the test suite runs under the simulator; we may need to make modifications to run on the hardware when we get it.

Task 2.3 Continue evaluation; document process and results in a report.

OS Prototype (Task 3): The third task involves the development of a prototype operating system that implements the design of the second phase and executes on a tagged-architecture processor.

Task 3.1 Continue improvement of simulators and deploy any versions of HW we receive from Cornell. We are considering looking now at the modification of the simulators to support evaluation of multi-core implementations of the processors and tagging schemes. We will conduct the evaluation this period to determine different possible approaches for this simulator and how it could impact test cases and the tagging model.

Task 3.2 Complete testing of modifications using simulator and Cornell HW. Received new hardware model Dec 22.

Task 3.3 None planned.

Security Tagged Multicore Processor

Single-core Architecture

Finish the evaluation for both tag compression scheme and the real-time system tagging.

Multi-core Architecture

We plan to start investigating possible options to add tagging modules to multi-core systems – shared vs. private.

Programming Model

N/A – Completed.

FPGA Prototype

We plan to work on implementing a precise exception support and look into a multi-core prototype.

CAD Infrastructure

N/A. There is no immediate task for CAD tools. Optionally, we plan to look into evaluating asynchronous FPGAs if possible.

Security Tagged Interconnect

Research Architectures

None planned.

Design Interconnect

Finish identifying component modifications and methods for isolating on-chip peripherals.

Interconnect Prototype

Start testing and incorporating interconnect design into Cornell Universities prototype.

Interconnect Security

Start identifying techniques to increase protections within the interconnect; i.e. eliminate man-in-the-middle attacks from other hardware components that can become compromised.

Project Cohesion

Continue coordinating interaction and collaboration between the University of Idaho and Cornell University with respect to the multi-core architectures that are now under development.